

Environmental factors promoting the evolution of recruitment strategies in swarms of foraging robots

Steven Van Essche¹, Eliseo Ferrante^{1†}, Ali Emre Turgut², Rinde Van Lon³,
Tom Holvoet³, and Tom Wenseleers¹

¹Laboratory of Socioecology and Social Evolution, KU Leuven

Naamsestraat 59, 3000 Leuven, Belgium; E-mail: eliseo.ferrante@bio.kuleuven.be

²Mechanical Engineering Department, Middle East Technical University, 06800, Ankara, Turkey

³Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium

Abstract: Swarm robotics has both an engineering as well as a scientific nature. From the engineering perspective, it studies how to design flexible, robust, and scalable collective behaviors to solve real-world problems in large, unstructured environments. From the scientific perspective, it is also useful to biologists to study the proximate mechanisms employed by social species to achieve the astonishing levels of collective organization that are often observed in nature. Evolutionary swarm robotics has a similar double nature. On the one hand, the use of evolutionary computation techniques is proposed as a solution to the design problem, that is, to decompose the collective-level goal into the local behaviors of the robots. On the other hand, studying evolutionary robotics scenarios can be very useful to biologists to understand the ultimate causes and factors that promote the evolution of specific types of collective organization in nature.

In this paper, our goal is to use an evolutionary swarm robotics scenario to answer questions related to the evolution of recruitment strategies in social insects. We consider a foraging scenario in which objects are distributed in the environment according to specific distributions. We show that the way food is distributed in the environment has a significant influence on whether and which recruitment strategies emerge through the evolutionary process. The results of this paper are therefore useful both to advance our understanding on the evolutionary causes of recruitment in biological systems, as well as to hint to engineers the requirements to evolve complex coordination strategies.

Keywords: swarm robotics; evolution of cooperation; signaling; recruitment; self-organization

1. INTRODUCTION

Applications such as search and destruction of cancer cells by nanobots, post-disaster search and rescue, and large environments exploration are particularly suited for large swarms of autonomous robots. Swarm robotics is the area of robotics that tries to employ a multitude of small, relatively simple robots to achieve intelligent collective-level behaviors [3, 25]. Swarm robotics draws inspiration from biology and by the incredible complex behavior exhibited by various species of animals and in particular the robustness, scalability, flexibility displayed by colonies of social insects [1, 3, 4, 6]. The design problem for swarm systems is hard [29], as a general way to translate a collective-level objective into the individual behaviors of the robots is missing. Current methodologies are divided into behavior-based (which are trial-and-error or seldom supported by a mathematical model) and automatic design. In the latter category, Evolutionary Robotics (henceforth ER) proposes a general methodology to synthesize individual-level behaviors given the collective objective expressed as the objective function of an evolutionary computation method [3, 11, 15, 17].

Swarm robotics can be also very useful to study biological systems. Owing to natural systems being black box entities, and mathematical models being too extreme simplifications of reality, embodied multi-robot simulations and robotic experiments have been used by biolo-

gists to study the collective behavior of social insects [14, 16, 28]. They provide a tool to test hypotheses on the proximate mechanisms underlying the self-organization of natural systems by directly implementing them on embodied agents or robots. Coupled with evolutionary computation methods, they become an even more powerful tool in the hands of biologists. When seen as a synthetic version of natural evolution, artificial evolution can be used to test hypotheses on the evolutionary conditions for the emergence of particular aspects of collective behaviors, such as communication [12, 13, 18, 30] and task specialization [10].

Understanding the factors that promote the evolution of highly organized collective behaviors is key to unravel the mysteries linked to the occurrence of the so-called major transitions in evolution [21], for example the transition from single to multi-cellular organisms and from solitary to eusocial insects, to name a few. It can also be very useful to understand what are the right conditions to automatically design highly organized collective behaviors that can be used to tackle tasks in complex environments [10].

In this paper, we aim at understanding the environmental factor leading to advanced forms of recruitment in societies of social insects. We focus on a foraging scenario, in which objects randomly scattered in the environment need to be retrieved by a swarm of robots. In social insect colonies, individuals do not collect selfishly during foraging, but rather signal explicitly to the others the lo-

† Eliseo Ferrante is the presenter of this paper.

cation of a significant food source, when found. In this way, more foragers are attracted to the food source, making the foraging process much more efficient, leading to direct benefits for the colony’s members [5, 20, 26].

We perform evolutionary experiments using the AR-GoS simulator [23], a realistic simulator where detailed models of robots are available. We show that the type of food distribution is a key factor that promotes the evolution of recruitment strategies in foraging. This is similar to a recent study on a similar scenario [19], which used less detailed multi-agent simulations and focused on pheromone strategies rather than those implementable on a swarm of robots. Our previous work has looked at the evolution of more general cooperation patterns such as foraging and task specialization [9, 10], but none have looked at recruitment strategies in foraging in the robotics community.

In our experiments, we compare the performance of recruitment strategies with those of solitary behaviors. The recruitment strategy that we employ is based on manually designed behavioral building blocks that are further optimized by a genetic algorithm. Although we have a few free parameters, this methodology still allows for the expression of a rich set of possible collective behaviors with different levels of collective organization.

Our results are relevant both for swarm robotics and for biology. From a biological point of view, our results give us a better understanding of why and how cooperation through recruitment evolved. In swarm robotics, our results show that the right environmental conditions, more than the fitness function, are very important to automatically synthesize collective behaviors with complex coordination strategies via ER.

2. PROBLEM DEFINITION

Analogously to a foraging scenario observed in ants or bees, in our scenario a swarm of n robots has to search and retrieve objects, which we will henceforth call *food*. Food is placed in random positions in the environment. We consider two types of food distribution. In uniform food distribution, the food is spread randomly covering the entire arena (see Figure 1a). In patched food distribution the food is concentrated in a restricted area of the environment, called patch, whose shape is square and whose central location is randomly chosen at the beginning of each experiment (see Figure 1b).

The goal of our research is to determine which foraging strategy is favored by evolution in each of these two environments. In the remaining of this section we will describe the simulated environment and robots that were used in the experiments. The experiments are carried using the ARGoS simulator [24], an open-source simulator able to simulate realistically complex experiments involving large swarms of robots. In Section 3 we will describe the methodology that we used to develop and evolve the collective behaviors.

2.1. The environment

The environments in our experiments are square areas delimited by walls through which the robots are unable to pass. Their shape is not relevant to the experiment and can be chosen arbitrarily. Different shapes would only change the time needed for a robot to find food. It is thus only important that the shape and size of the environment stays consistent throughout our experiments, in order to perform meaningful comparisons between strategies.

The robots are initially placed near a gray square area called *nest* located in the middle of the arena. Robots can easily return to the nest thanks to a light that is placed above the nest. They can detect when they are in the nest by using their ground sensor that can measure the color of the area beneath them. The nest is the area where the food must be returned. The size of the nest is big enough to accommodate multiple robots returning food at the same time.

Food items are circular and can be distinguished by the robots by detecting its black color using the ground sensors. Food is never placed in the nest nor too close to the nest because otherwise there would be no need for foraging. Throughout the experiments, the average distance to all the food objects is kept constant. This means that the distance from the nest to the center of a patch (in the patched environment) is determined by the average distance from the nest to all uniformly distributed food objects (in the uniform environment). We do this in order not to bias foraging towards a specific kind of environment, and to have a fair comparison.

Food also has a renewal rate which determines at what rate it reappears in the environment. For simplicity, in our experiments, one food item will reappear in the environment in a random location (uniformly or in the patch) once an other item has been retrieved to the nest, that is, we consider the highest possible rate. We believe that different renewal rates would possibly lead to different recruitment strategies, but this study is left to future investigation. The total number of food items is saturated to 50. For a summary of the parameters characterizing the environment and on their values, refer to Table 1. The swarm’s performance is measured by its collection rate and the total amount of food it is able to collect.

Parameter description	Value
Size of the environment	10 units
Size of the nest	2 units
Food distribution type	patched or uniform
Time steps to rep	1
Max. number of food items	50
Radius of a food item	0.1 units

Table 1 Environment parameters used in the experiments.

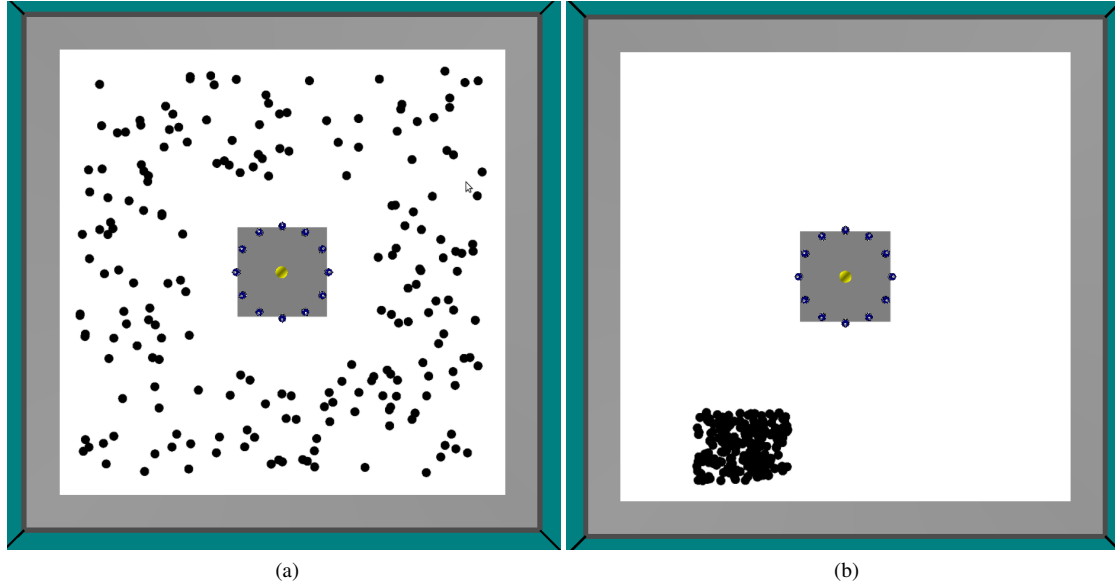


Fig. 1 The two types of environments used in our experiments: (a) uniform versus (b) patched environment.

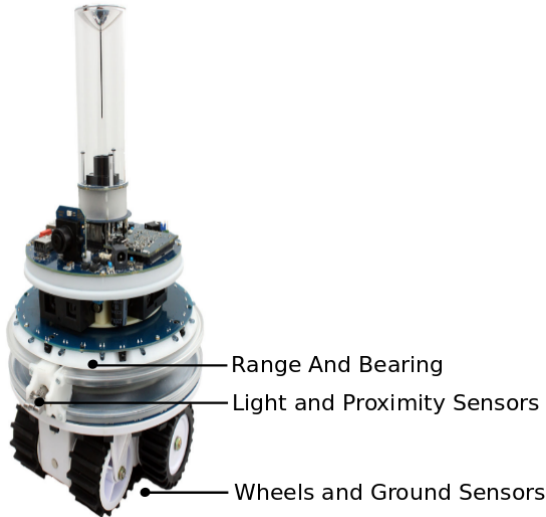


Fig. 2 The physical marXBot with the sensors/actuators that we used in our experiments.

2.2. Robots

The robots involved in our experiments are simulated versions of the marXbot [2], a differential-drive robot with various actuators and sensors. This robot is depicted in fig.2. Among the plethora of sensors and actuators available to this robot, we only use those depicted in the picture. The range and bearing sensor, which allows robot-to-robot sensing and local communication, is used to avoid other robots and for the signaling behavior. The light and proximity sensor, which allows robot to detect lights and to sense objects in near proximity, is used to detect the light placed above the nest and to avoid walls. The ground sensors are used to detect the color of the ground. Finally, the wheels are used to navigate in a differential-drive fashion. In Section 3 we will better describe how the sensors are used.

3. METHODS

The behavior exhibited by our simulated robots is based on a modular architecture [8].

3.1. Behavioral building blocks

We implemented the following low-level behavioral building blocks:

Random Walk The robot moves in a straight line for a random amount of time and then changes direction randomly (uses the wheel actuators and no sensors).

Phototaxis The robot moves towards to direction corresponding to the highest light intensity (uses the light sensor and the wheels actuators).

Anti-Phototaxis The robot moves away from the direction corresponding to the highest light intensity (uses the light sensor and the wheels actuators).

Observe Ground The robot senses the color of the area beneath it to distinguish between nest, regular environment and food (uses the ground sensors).

Obstacle Avoidance The robot senses other robots and walls and changes its path to avoid collisions with them (uses the proximity sensors, the range and bearing actuators and sensors, and the wheels actuators).

Signal The robot sends a signal to other robots in local proximity (uses the range and bearing actuators and sensors).

Follow signal The robot receives signals from other robots in local proximity, and moves towards the direction of the strongest signal perceived (uses the range and bearing actuators and sensors and the wheels actuators).

These behaviors are combined into higher-level behaviors that are associated with robot states. The finite state machine used by each robot is determined by their role. Swarm robots can have one among three possible roles: *recruiter*, *recruitee* or *solitary*. The intuitive idea behind these roles is the following:

Solitary The robot performs a random walk in the envi-

ronment, immediately carrying off any food that it finds to the nest. It does not signal other robots and ignores other robots' signals completely.

Recruiter The robot performs a random walk in the environment. When it has found food, it sends a signal to the other robots for a given duration.

Recruitee The robot performs a random walk in the environment, listening to signals of recruiters. When a signal is detected, a recruitee moves towards the source of this signal. Once very close to the signal, it starts exploring that area for food. If food is found, it is immediately carried to the nest.

Each of these roles is implemented by a finite state machine (henceforth FSM) that uses the previously defined low-level behaviors. We will now take a closer look at their finite state machines:

3.1.1. Solitary

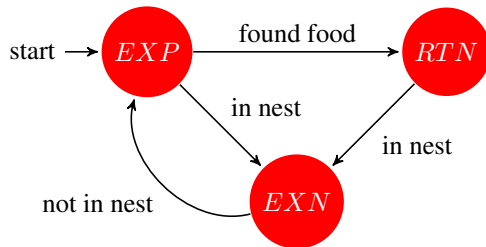


Fig. 3 FSM of a solitary robot.

The solitary robot's FSM is a fairly simple one and is depicted in Figure 3. It consists of three states:

Explore (EXP): The robot performs the **Random Walk** behavior to explore its environment while using **Obstacle Avoidance** to avoid any obstacles.

Return to nest (RTN): The robot uses **Phototaxis** and **Obstacle Avoidance** to find its way back to the nest with the food object that it has picked up.

Exit nest (EXN): The robot uses **Anti-Phototaxis** and **Obstacle Avoidance** to leave the nest as quickly as possible.

The robot starts in the *Explore* state until it finds a food object. It then proceeds with picking up the food object and switching its state to the *Return to nest* state. Once it has arrived back to the nest, it drops its food, triggering the *Exit nest* state. When the robot has left the nest, the whole process starts over. The *Exit nest* state is also triggered whenever the robot accidentally enters the nest without food.

3.1.2. Recruiter

The recruiter robot is similar to the solitary robot, but is specialized in recruiting other robots to any food source it finds. Its FSM can be seen in fig. 4. This FSM introduces one new state:

Signal food (SIG): The robot stops moving and uses the **Signal** behavior to broadcast a signal to nearby robots.

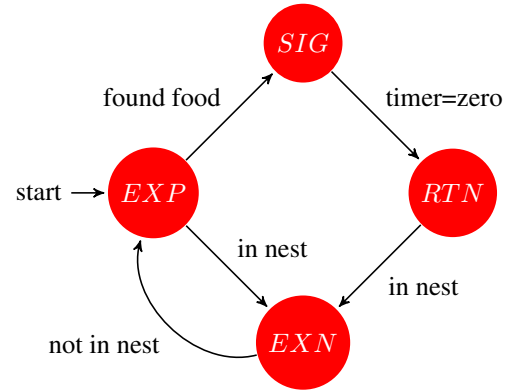


Fig. 4 FSM of a recruiter robot.

The signal can be only perceived by robots in close proximity (in our experiments, 4 meters, determined by the default sensing range of the range and bearing sensor of the marXbot).

This robot's behavior is similar to the solitary robot's behavior, but instead of switching to the *Return to nest* state when food is found, it switches to the *Signal food* state. When it enters this state it starts a timer. When that timer reaches zero, the robot stops signaling and enters the *Return to nest* state.

3.1.3. Recrutee

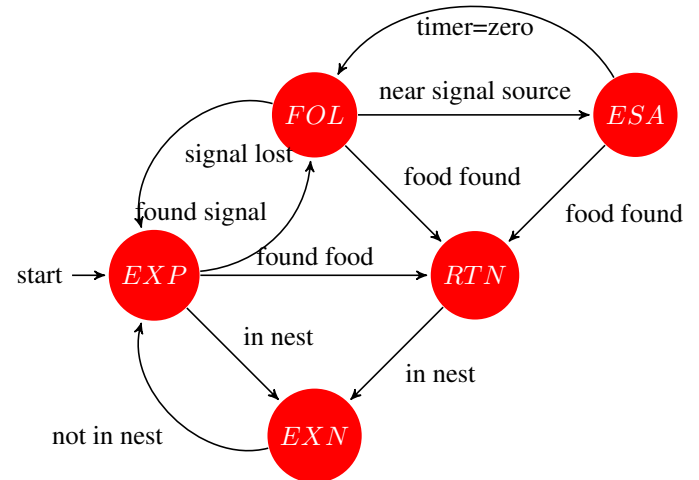


Fig. 5 FSM of a recruitee robot.

The recruitee is specialized in finding signals and following them. Its FSM can be seen in fig. 5. Its FSM introduces the following new states:

Follow signal (FOL): The robot detects a signal and moves towards the direction of the signal's source using the **Follow signal** behavior and the **Obstacle Avoidance** behavior to avoid collisions with other robots and objects.

Explore signal area (ESA): The robot explores the area around the signal (in a similar way as in the *Explore* state) but here it also keeps track of how long it has been exploring the signal source's area.



Fig. 6 The results of using a solitary versus a recruitment strategy in a uniform and patched environment.

Instead of simply exploring the environment, this robot also listens for signals of recruiters. Whenever it picks up such a signal, it switches to the *Follow signal* state. At this point, the robot can either lose the signal, find its source, or encounter food on its way to the signal's source. Finding food immediately triggers the *Return to nest* state, while losing the signal results in the robot reverting back to the *Explore* state. When a robot succeeds in reaching a signal's source (i.e. entering within a specific range of the signal source), it switches to the *Explore signal area* state. In this state, the robot can either find food, thus switching to the *Return to nest* state, or give up after a while and return to the *Follow signal* state. The rationale behind this behavior is that the robot might have wandered off too far from the signal area, allowing it to once more return to the signaled food source.

3.2. Evolutionary method

As it can be deduced from the previous section, a number of parameters need to be chosen for the recruitment strategy. These are:

Robot distribution

This determines the proportion of robots engaged in the solitary, recruiter and recruitee strategies, respectively.

Exploration time

This is the time spent by the recruitee in the *Explore signal area* state.

Signaling time

This is the time spent by the recruiter in the *Signal food* state.

Signal closeness range

This is how close a robot will try to be to a signal source before it starts exploring the signal area.

Despite the number of parameters not being huge, we chose them in a way to maximize the variety of collective dynamics that can be in principle achieved. The first parameter, the robot distribution, is clearly key in determining the collective dynamics and the performance of the swarm. In principle, the problem can be seen as an evolutionary game played by robots, each of which could

be playing one of the three strategies: solitary, recruiter, or recruitee. This would correspond to the same type of analysis as the one in [10] and in an ongoing work in which we modeled such dynamics using replicators equations. Such analysis is however beyond the scope of the current paper. Furthermore, the other parameters have also a substantial impact on the collective dynamics and performance. For instance, the signaling time determines the cost of signaling, that is, how much of its own fitness a recruiter robot is willing to sacrifice to benefit the colony in the attempt to increase the overall colony efficiency. Similarly, the exploration time and the signal closeness range are used to fine-tune the recruitee behavior and the recruiter-recruitee behavioral interactions.

Each robot in a swarm of n robots is executing the same controller. The genotype representations of the controller is simply a tuple of integers encoding the parameters described above. We evaluate the fitness function by measuring the amount of food the swarm is able to collect in the allotted simulation time. Each chromosome is evaluated 20 times using a different seed each time to generate a different environment with the same food distribution (uniform or patched). The fitness is the mean fitness over 20 runs. We perform selection following a fitness-proportional scheme, or roulette-wheel selection. We define a selection rate that determines how many individuals are allowed to reproduce, but also how many are substituted to create the next generation. Our selection rate is 90%, which also implies an elitism of 10% (i.e. the 10% best individuals are preserved for the next generation). We used a population size for the GA of 20 individuals, 30 generations, a mutation probability of 5% and crossover probability of 90%. Each experiment was ran for 50 000 simulated time steps, which was empirically determined as sufficient for the swarms to converge on a steady collection rate.

4. EXPERIMENTS

In all our experiments, the swarm size was selected to be $n = 12$. We chose this number as it was high enough

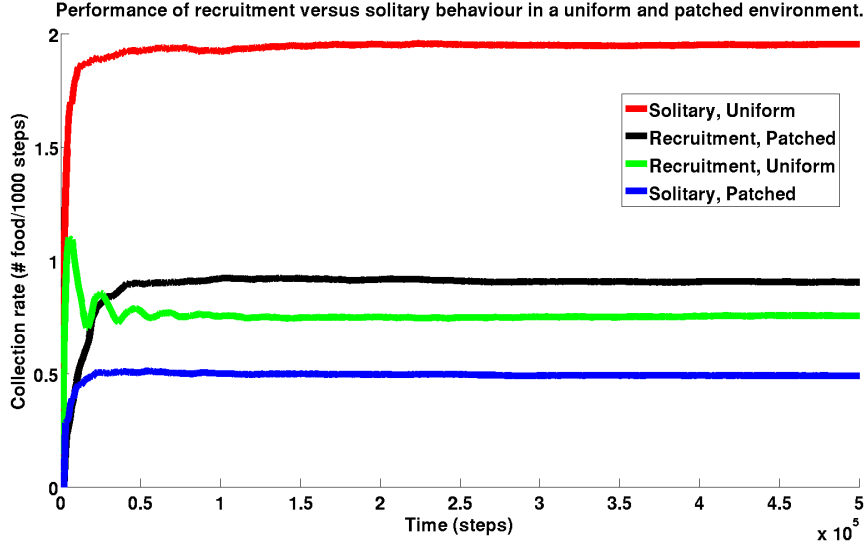


Fig. 7 The collection rates achieved in the four different experiments.

to obtain rich collective dynamics and low enough to be able to execute enough experiments with the allotted computational resources. To avoid situations in which robots would initially all head off in the same directory by chance, we’ve initially placed all robots on a circle around the center of the nest facing outwards (see fig. 1). This way the time needed by the swarm to converge on their constant collection rate is reduced.

We first executed two evolutionary runs, one in the uniform environment and one in the patched environment. The final evolved controller in the uniform environment resulted in a swarm that does not use recruitment. In other words, the resulting swarm only contained solitary robots. However, in the patched environment, the final evolved controller corresponds to a swarm that does employ recruitment. The characteristics and corresponding parameters of this swarm are given in Table 2. For videos demonstrating the resulting controllers, refer to [7].

Parameter	Value
# Solitaires	0
# Recruiters	4
# Recruits	8
Exploration time	300 time steps
Signaling time	15 000 time steps
Signal closeness range	20 units

Table 2 GA results for the patched environment.

The results suggest that the evolutionary process leads to a swarm that has a 1:2 ratio of recruiters versus recruits. We can also see that once a recruiter finds a patch, it stays there for a very long time. One might have expected that it would even be better for the recruiter to stay in the same spot for the entire experiment since there is only one patch. This is however not true for a number of reasons. If a lot of recruiters find the patch and stay there indefinitely, they might actually block the patch as the robots will avoid coming too close to each other. Fur-

thermore, it comes to the cost of not contributing to the overall collection task, as described above.

These results show that recruitment strategies do not readily evolve in environments that are too uniform, while they do evolve in presence of heterogeneous environments with features that can be exploited by individuals. In the case of the current scenarios, these features are represented by information that can be exploited and communicated by the robots (food location). In another scenario that we considered in an earlier study [10], we reached a similar conclusions. In that study, a complex coordination strategy that we referred to as task specialization emerged only in presence of non-uniform environments, in that case represented by a non-flat arena where a slope could be exploited to transfer food items more economically, which resulted in a task-partitioning behavior. We speculate that there might be a general principle governing the emergence of complex coordination strategies, whereby information transfer maximization could be its driving factor.

These two controllers we then evaluated in four set of experiments:

Uniform → *Uniform* We evaluated a robot swarm evolved in a uniform environment (fully solitary strategy) into a uniform environment.

Uniform → *Patched* We evaluated a robot swarm evolved in a uniform environment (fully solitary strategy) into a patched environment.

Patched → *Uniform* We evaluated a robot swarm evolved in a patched environment (recruitment-recruitee strategy) into a uniform environment.

Patched → *Patched* We evaluated a robot swarm evolved in a patched environment (recruitment-recruitee strategy) into a patched environment.

Each experiment set consists of 60 evaluation runs.

The results are shown in Figure 6. The results clearly show that each evolved behavior was optimal for the environment in which it was evolved, and did not transfer very well to a different environment. We also see that

performance in the uniform environment are in general much higher than in the patched environment. This is due to the natural increased difficulty of the task in the patched environment: In the uniform environment, food can be found anywhere, and there is no need to perform search. In the patched environment, food needs first to be searched, before being able to retrieve it effectively. This is also reflected in the increased standard deviation of the performance of the recruitment-based behavior in the patched environment: the performance of a given run will depend strongly on the initial time needed to find the food patch by the first robot(s).

Finally, Figure 7 reports a comparison of the the average performance in the four experimental evaluations, plotted as a function of time. The figure confirms that the best performance are obtained using a solitary strategy in a uniform environment, confirming the intrinsic simplicity of such a scenario. We also notice that a recruitment strategy in the uniform environment has an initial performance boost, which then is attenuated: this behavior corresponds to the fact that the signaling strategy is not giving an effect in the initial moment of the simulation, and as such recruits are behaving like solitary robots. Once recruitment starts, this will have a negative effect on the swarm performance due to the absence of positional information that can be exploited. We also notice the presence of fluctuating peaks after this initial peak, which can be explained by a "synchronization" effect of recruiters stopping signaling and bringing food together to the nest. This initial effect is then dampened once recruiters stop being synchronized. Besides this, Figure 7 confirms the general message, being that each controller performs the best in the environment where it first evolved.

Overall, our results show that specific environmental conditions are critical for the emergence of complex coordination strategies. In our setting, these strategies did not transfer well to other environments, but this immediately raises questions on what would then be the right environmental conditions required for the emergence of flexible coordination strategies, that is, strategies that would automatically switch from solitary to recruitment-type of behaviors depending on the environmental situations. Determining such environment is object to future work, and we will be determined by looking at the environment present in real biological systems. We speculate that a power-law type of distribution of food items in the environment might be enough for evolving an adaptive foraging strategy [19].

5. CONCLUSION

In this paper, we showed how environmental factors play a key role in the evolution of complex coordination strategies, such as recruitment and signaling. To support this hypothesis, we performed evolutionary experiments using realistic swarm robotics simulations. Our methodology relied on the presence of existing behavioral-building blocks, that is, behavioral primitives able to carry out basic tasks such moving towards

the light, avoiding obstacles, etc . . . , in a similar fashion as it is believed to happen through natural evolution [22]. Using a simple GA, we were able to evolve a rich variety of collective behaviors to cope with a foraging task in two environments: a uniform environment where food items are placed uniformly at random positions, and a patched environment where items are placed in bounded patches at given random locations.

Our results show that a complex recruitment strategy could only evolve in the patched environment, while a solitary strategy evolved in the uniform environment. Furthermore, each strategy was shown to be performing well only in its original environment where it first evolved. This results prompt the need to answer the question of what would be the right environment that would favor a flexible foraging strategy, able to switch from solitary to recruitment as needed.

This study is important but only the first milestone in our research agenda. Our next step is to investigate the role of genetic structure on the evolution of signaling and recruitment behaviors. In evolutionary biology, it is well known that explicit signaling should be favored in groups composed of genetically related individuals, whereby sharing information about food benefits the entire colony and hence produces an inclusive fitness effect, but less so in groups of non-related individuals, whereby different strategies such as eavesdropping emerge [27]. Future work aims at using a similar scenario to the one considered here with a different evolutionary framework that would allow the study of the effect of genetic relatedness on the evolution of different recruitment mechanisms. Furthermore, we plan to develop a theoretical model able to predict when and which type of recruitment model can be favored by evolution under different environmental situations. Finally, our ultimate goal is develop a more general, rather than case-specific, understanding of what are the factors leading to the emergence of complex coordination strategies, that encompasses not only the scenario considered here but also the one we studied in [10] and other collective behaviors observable in natural system. This way, we can not only shed more light on the evolutionary causes in biology, but also develop the right tools and recommendations for the automatic design of collaborative strategies for artificial swarms of robots.

REFERENCES

- [1] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence*. Oxford, 1999.
- [2] Michael Bonani, Valentin Longchamp, Stéphane Magnenat, Philippe Régnier, Daniel Burnier, Gilles Roulet, Florian Vaussard, Hannes Bleuler, and Francesco Mondada. The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4187–4193. IEEE, 2010.
- [3] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review

- from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.
- [4] Scott Camazine. *Self-organization in biological systems*. Princeton University Press, 2003.
 - [5] Ruth Chadab and Carl W Rettenmeyer. Mass recruitment by army ants. *Science*, 188(4193):1124–1125, 1975.
 - [6] Marco Dorigo and Mauro Birattari. Swarm intelligence. *Scholarpedia*, 2(9):1462, 2007.
 - [7] Steven Van Essche, Eliseo Ferrante, Ali Emre Turgut, Rinde Van Lon, Tom Holvoet, and Tom Wenseleers. Environmental factors promoting the evolution of recruitment strategies in swarms of foraging robots. Supplementary materials, 2015. <http://bio.kuleuven.be/ento/ferrante/FerranteSup2015-001/index.html>.
 - [8] E. Ferrante. A control architecture for a heterogeneous swarm of robots: The design of a modular behavior-based architecture, 2009. MAS report - Université Libre de Bruxelles.
 - [9] E. Ferrante, E. Duéñez Guzmán, A. E. Turgut, and T. Wenseleers. Geswarm: Grammatical evolution for the automatic synthesis of collective behaviors in swarm robotics. In *Proceedings of the fifteenth international conference on Genetic and evolutionary computation conference companion*, pages 17–24. ACM, New York, NY, 2013.
 - [10] E. Ferrante, A.E. Turgut, E. Duéñez-Guzmán, M. Dorigo, and T. Wenseleers. Evolution of self-organized task specialization in robot swarms. *PLOS Computational Biology*, 11(8):e1004273, 2015.
 - [11] D Floreano, S Nolfi, et al. Evolutionary robotics. the biology, intelligence, and technology of self-organizing machines. Technical report, MIT Press, 2001.
 - [12] Dario Floreano and Laurent Keller. Evolution of adaptive behaviour in robots by means of darwinian selection. *PLoS Biol*, 8(1):e1000292, 01 2010.
 - [13] Dario Floreano, Sara Mitri, Stéphane Magnenat, and Laurent Keller. Evolutionary Conditions for the Emergence of Communication in Robots. *Current Biology*, 17:514–519, 2007. communication.
 - [14] Simon Garnier, Maud Combe, Christian Jost, and Guy Theraulaz. Do ants need to estimate the geometrical properties of trail bifurcations to find an efficient route? a swarm robotics test bed. *PLoS Comput Biol*, 9(3):e1002903, 03 2013.
 - [15] David Edward Goldberg et al. *Genetic algorithms in search, optimization, and machine learning*, volume 412. Addison-wesley Reading Menlo Park, 1989.
 - [16] J. Halloy, G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tche, I. Sad, V. Durier, S. Canonge, J. M. Am, C. Detrain, N. Correll, A. Martinoli, F. Mondada, R. Siegwart, and J. L. Deneubourg. Social integration of robots into groups of cockroaches to control self-organized choices. *Science*, 318(5853):1155–1158, 2007.
 - [17] John H Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
 - [18] M. J. Krieger, J. B. Billeter, and L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406(6799):992–5, 2000.
 - [19] Kenneth Letendre and Melanie E. Moses. Synergy in ant foraging strategies: Memory and communication alone and in combination. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO '13*, pages 41–48, New York, NY, USA, 2013. ACM.
 - [20] Maurice Materlinck. *The Life of the Bee*. Wildside Press LLC, 2008.
 - [21] J. Maynard-Smith and E. Száthmary. *The Major transitions in evolution*. W.H.Freeman, 1995.
 - [22] R. E. Page Jr, T. A. Linksvayer, and G. V. Amdam. *Social life from solitary regulatory networks: a new paradigm for insect sociality*. Harvard University Press, Cambridge, Massachusetts, 2009. n/a.
 - [23] C. Pinciroli, V. Trianni, R. O’Grady, G.i Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295, 2012.
 - [24] Carlo Pinciroli, Vito Trianni, Rehan OGrady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, et al. Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm intelligence*, 6(4):271–295, 2012.
 - [25] Erol Şahin. Swarm robotics: From sources of inspiration to domains of application. In *Swarm robotics*, pages 10–20. Springer, 2005.
 - [26] T.D. Seeley. *Honeybee Democracy*. Princeton University Press, 2010.
 - [27] D.J.T. Sumpter. *Collective Animal Behavior*. Princeton University Press, 2010.
 - [28] Guy Theraulaz, Simon Goss, Jacques Gervet, and Jean-Louis Deneubourg. Task differentiation in polistes wasp colonies: A model for self-organizing groups of robots. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats*, pages 346–355, Cambridge, MA, USA, 1990. MIT Press.
 - [29] Vito Trianni and Stefano Nolfi. Engineering the evolution of self-organizing behaviors in swarm robotics: A case study. *Artificial life*, 17(3):183–202, 2011.
 - [30] Steffen Wischmann, Dario Floreano, and Laurent Keller. Historical contingency affects signaling strategies and competitive abilities in evolving populations of simulated robots. *Proceedings of the National Academy of Sciences*, 109(3):864–868, 2012.